PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Luc Van Brabant	Technology Center: 2100
Serial No.: 10/748,008 Confirm 6494	Group Art Unit: 2439
Filed: 12/30/2003	Examiner: Wang, Harris C.
For: ON-ACCESS AND ON-DEMAND DISTRIBUTED VIRUS SCANNING	Atty. Dkt. No.: 10830.0103.NP

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief is in support of Appellant's Notice of Appeal (Second Reinstatement) filed Oct. 21, 2009 in reply to the Final Official Action of July 21, 2009. Please apply the appeal brief fee of $510 previously paid on Feb. 26, 2008, and charge the additional fee of $30 (due to the FY 2009 fee increase) to EMC Corporation Deposit Account No. 05-0889. Please find submitted herewith a Fee Transmittal For FY 2009 form authorizing the charge of the additional fee of $30 and the charge of any other required fee for this Appeal Brief to EMC Corporation Deposit Account No. 05-0889.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 014859 Frame 0273.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1-33 have been presented for examination.

Claims 1-5, 8-10, and 16-21 have been cancelled.

Claims 29, 31, 32, and 34 have been allowed.

Claims 6, 7, 11-15, 22-28, 30, and 33 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

An Amendment in Reply to Final Official Action was filed on Oct. 21, 2009, and according to an Advisory Action dated Nov. 3, 2009, this Amendment in Reply to Final Official Action was entered for purposes of Appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The appellant's invention relates generally to "on-access" virus scanning and "on-demand" virus scanning. "On-access" virus scanning occurs when a specified trigger occurs, such as when a user accesses a file marked "unchecked." (Appellant's specification, page 4, lines 7-8.) "On-demand" virus scanning is typically scheduled when a new virus is discovered, when new unchecked files are migrated to a file server, or prior to archiving or backing-up unchecked files. (Appellant's specification, page 4, lines 8-10.)

The appellant's invention of independent claim 6 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 4, line 21-23; page 9 lines 3-7). The method of claim 6 includes combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue (63 in FIG. 3), and distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 4, line 23 to page 5, line 2; page 13 lines 13-15; page 13 lines 1-7). The method of claim 6 further includes grouping the on-demand anti-virus scan requests into chunks (67 in FIG. 3), each of the chunks including multiple ones of the on-demand anti-virus scan requests (steps 108-109 in FIG. 7), and placing the chunks onto the virus scan request queue (step 110 in FIG. 7; step 126 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 13 lines 13-17 and 20-22; page 16, lines 18-22; page 17, lines 12-14.)

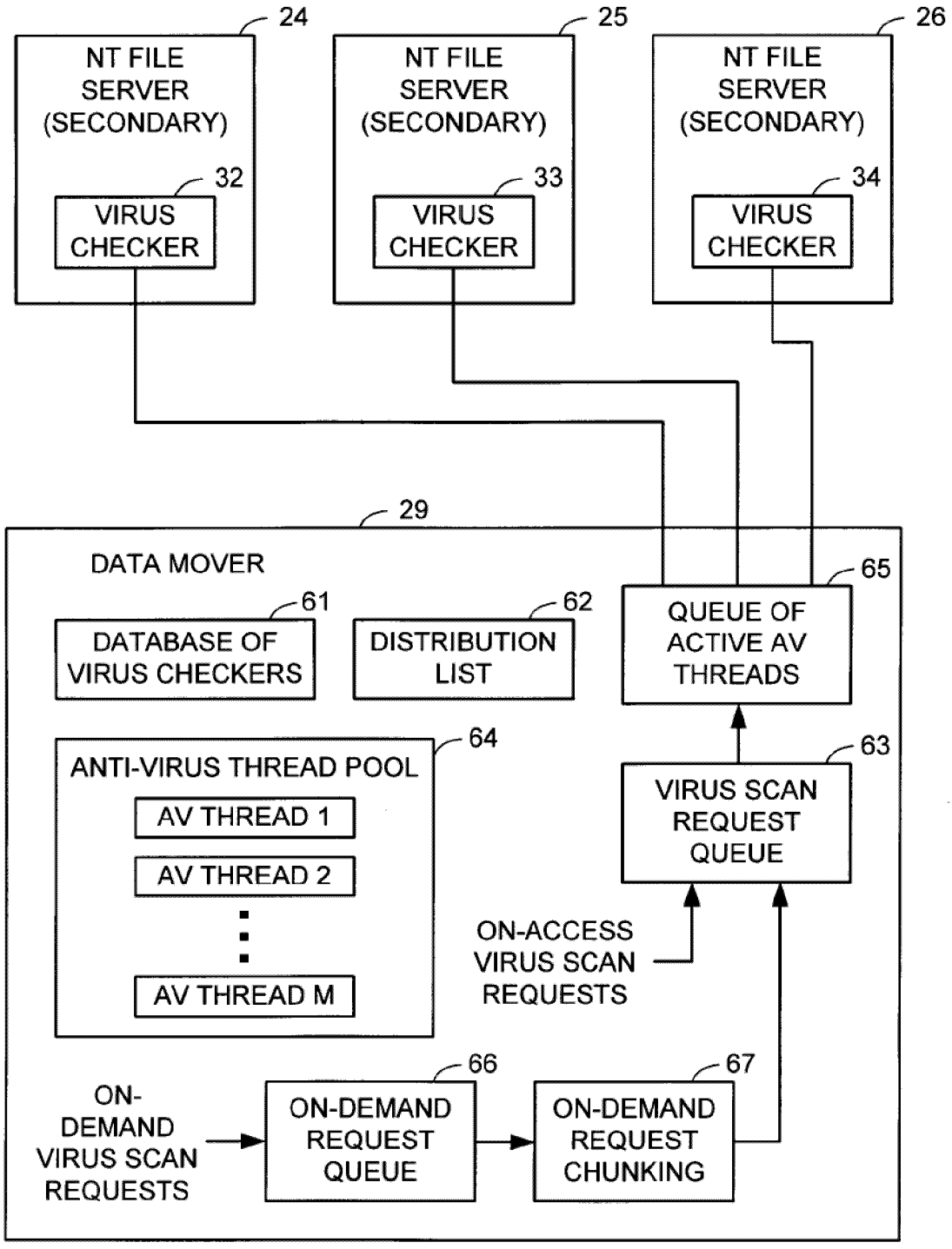
24
NT FILE SERVER (SECONDARY)
32
VIRUS CHECKER
25
NT FILE SERVER (SECONDARY)
33
VIRUS CHECKER
26
NT FILE SERVER (SECONDARY)
34
VIRUS CHECKER
29
DATA MOVER
61
DATABASE OF VIRUS CHECKERS
62
DISTRIBUTION LIST
65
QUEUE OF ACTIVE AV THREADS
64
ANTI-VIRUS THREAD POOL
AV THREAD 1
AV THREAD 2
AV THREAD M
63
VIRUS SCAN REQUEST QUEUE
ON-ACCESS VIRUS SCAN REQUESTS
66
ON-DEMAND REQUEST QUEUE
67
ON-DEMAND REQUEST CHUNKING
ON-DEMAND VIRUS SCAN REQUESTS

FIG. 3

For example, the on-demand anti-virus scan requests and the on-access anti-virus scan requests are distributed from the virus scan request queue to the virus checkers by AV threads in a pool (64 in FIG. 3), and each AV thread is programmed as shown in steps 71 to 83 of FIGS. 4 and 5 to send the request from the head of the virus scan request queue (step 76 in FIG. 4) to a particular one of the virus checkers (step 83 in FIG. 5) assigned by a distribution list (62 in FIG. 3), as described in appellant's specification on page 13 lines 1-4 and page 13 line 23 to page 14 line 21.

The appellant's invention of independent claim 11 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3). (Appellant's specification, page 5, lines 3-4; page 9 lines 3-7.) The method includes distributing on-demand anti-virus scan requests and on-access anti-virus scan requests to the virus checkers so that the virus checkers perform on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 5, lines 4-7.) The method further includes grouping the on-demand anti-virus scan requests into chunks of multiple ones of the on-demand anti-virus scan requests (67 in FIG. 3; steps 108-109 in FIG. 7), and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers. (Appellant's specification, page 5, lines 7-10; page 13 lines 13-17 and 20-22; page 16, lines 18-22.) The method further includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks (step 110 in FIG. 7; steps 121, 122, and 123 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 16, lines 18-22; page 17, line 12 to page 18, line 3.)

The appellant's invention of independent claim 12 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 5, lines 11-13; page 9 lines 3-7.) The method includes combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue (63 in FIG. 3), and a pool of threads (64 in FIG. 3) distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 5, lines 13-16.) Each anti-virus scan request on the virus scan request queue is serviced by a respective one of the threads in the pool of threads. (Appellant's specification, page 5, lines 17-18.) The method further includes grouping the on-demand anti-virus scan requests into chunks of multiple ones of the on-demand anti-virus scan requests (67 in FIG. 3; steps 108-109 in FIG. 7), and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold (TH1 in step 124 of FIG 8), and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing the chunk on the virus scan request queue (step 126 in FIG. 8). (Appellant's specification, page 5, lines 18-23; page 13 lines 13-17; page 17 line 21 to page 18 line 3.)

The appellant's invention of independent claim 22 provides a virus checking system including a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, a virus scan request queue (63 in FIG. 3), and at least one processor (29 in FIG. 3) coupled to the virus checkers and the virus scan

request queue for sending virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 6, lines 2-6; page 9 lines 3-7; page 8 line 22 to page 9 line 1.) The at least one processor is programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue, and for distributing the on-demand anti-virus scan requests and the on-access virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 6, lines 6-10.) The at least one processor is further programmed for grouping the on-demand anti-virus scan requests into chunks (67 in FIG. 3), each of the chunks including multiple ones of the on-demand anti-virus scan requests (steps 108-109 in FIG. 7), and placing the chunks onto the virus scan request queue (step 110 in FIG. 7; step 126 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 13 lines 13-17 and 20-22; page 16, lines 18-22; page 17, lines 12-14.)

The appellant's invention of independent claim 24 provides a virus checking system including a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, and a file server (27 in FIG. 1) coupled to the virus checkers for sending virus checking requests from the file server to the virus checkers. (Appellant's specification, page 6, lines 11-15; page 8 lines 8-11 and 17-18; page 9 lines 3-7.) The file server includes a virus scan request queue (63 in FIG. 3). (Appellant's specification, page 6, line 15; page 12 lines 13-15; page 13, lines 2-7.) The file server is programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue, and for executing multiple threads (64 in FIG. 3) for distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests

from the virus scan request queue to the virus checkers (steps 71 to 83 of FIGS. 4 and 5). (Appellant's specification, page 6, lines 15-19; page 13 lines 2-17; page 13 line 23 to page 14 line 21.) Each anti-virus scan request on the virus scan request queue is serviced by a respective one of the threads in the pool of threads (steps 71 to 83 of FIGS. 4 and 5). (Appellant's specification, page 6, lines 19-21; page 13 line 23 to page 14 line 21.) The file server is further programmed for grouping the on-demand anti-virus scan requests into chunks (67 in FIG. 3; steps 108-109 in FIG. 7) of multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue. (Appellant's specification, page 6, lines 21-23; page 13 lines 13-17 and 20-22; page 16, lines 18-22.)

None of appellant's claims contain any "means plus function" or "step plus function" as permitted by 35 U.S.C. 112, sixth paragraph.

Appellant's dependent claims 7, 15, 23, and 28 further define inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks (step 110 in FIG. 7; steps 121, 122, and 123 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 16, lines 18-22; page 17, line 12 to page 18, line 3.)

Appellant's dependent claims 30 and 33 further define that the chunks have a size equal to a certain number of on-demand anti-virus scan requests, and the threshold is one-half of the size. (Appellant's specification, page 13 lines 18-22.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 30 and 33 are indefinite under 35 U.S.C. 112, second paragraph, for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

2. Whether claim 6 is unpatentable under 35 U.S.C. 102(e) as being anticipated by Smithson et al. U.S. 6,802,012.

3. Whether claims 11-15 and 22-28 are unpatentable under 35 U.S.C. 102(e) as being anticipated by McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002.

4. Whether claims 7, 12, and 24 are unpatentable under 35 U.S.C. 103(a) as being obvious from Smithson et al. U.S. 6,802,012 in view of McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002.

5. Whether claims 30 and 33 are unpatentable over McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002, in view of in view of Edwards U.S. Patent 7,188,367.

VII. ARGUMENT

1. Claims 30 and 33 are not indefinite under 35 U.S.C. 112, second paragraph, and particularly point out and distinctly claim the subject matter which applicant regards as the invention.

On page 4 of the final Official Action, claims 30 and 33 were rejected under 35 U.S.C. 112, second paragraph, for being indefinite with respect to the term "about one-half." In reply, in appellant's Amendment in Response to Final Official Action filed Oct. 21, 2009, claims 30 and 33 were amended to remove the word "about." The Advisory Action dated Nov. 3, 2009, indicates in paragraph 7 that this amendment will be entered for purposes of appeal. Therefore there is no longer a basis for maintaining the rejection of claims 30 and 33 under 35 U.S.C. 112, second paragraph.

2. Claim 6 is patentable under 35 U.S.C. 102(e) and is not anticipated by Smithson et al. U.S. 6,802,012.

Anticipation is established only when a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. RCA Corp. v. Appl. Dig Data Sys.. Inc., 730 F.2d 1440, 1444 (Fed. Cir. 1984); WL. Gore & Assocs., Inc. v, Garlock, Inc., 721 F.2d 1540, 1554 (Fed Cir. 1983). The elements must be arranged in

the reference as in the claims under review, although this is not an ipsis verbis test. In re Bond, 910 F.2d 831,15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation).

With respect to appellant's claim 6, Smithson does not show the recited claim elements "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue." (Emphasis added.) Thus, appellant's claim 6 recites a "grouping" operation distinct from the "placing" operation, and what is placed by the "placing" operation are "the chunks" produced by the "grouping" operation. The placement of chunks of plural on-demand scan requests on the virus scan request queue, instead of individual on-demand scan requests, is shown in appellant's FIG. 7, steps 108, 109, and 110, as described in appellant's specification, page 16 lines 4-8 and 18-23.

Smithson discloses a method of operating a computer for on-demand anti-virus scanning and on-access virus scanning of computer files. (Smithson, Abstract.) On-demand anti-virus scan requests and on-access anti-virus scan requests are combined in a virus scan request queue. Smithson's virus scan request queue is called a "pending list" in step 18 of Smithson's FIG. 2. This "pending list" is shown in Smithson's FIG. 3 for the case of on-access requests rather than on-demand accesses. (Smithson, col. 5 lines 5-22.) The scan requests in the pending list, however, are not distributed to a plurality of virus checkers. Instead, the scan requests in the pending list are sequentially selected and serviced one at a time in step 24 of Smithson's FIG. 4. (Smithson, col. 5, lines 23-30.) Therefore, at any given time that the Scan Engine 34 of

Smithson's FIG. 5 is performing an anti-virus scan, either an on-demand anti-virus scan or an on-access anti-virus scan is being performed by the Scan Engine 34. (Smithson, col. 5, lines 46-54.) Thus, a person of ordinary skill recognizes that the Scan Engine 34 in Smithson FIG. 5 is a virus checker, but there are not, in Smithson FIG. 5, "a plurality of virus checkers for on-demand anti-virus scanning concurrent with one-access anti-virus scanning ... by distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers; ..." as recited in appellant's claim 6.

With respect to claim 6, page 6 of the final Official Action says: "Figure 2 of Smithson teaches placing on-demand scan requests into a queue, Figure 3 shows the 'chunks'." However, Smithson does not disclose grouping on-demand scan requests into these chunks, and placing these chunks onto the virus scan request queue. Instead, Figure 2 of Smithson shows that one on-demand or on-access scan request is received in step 10 and then written to the virus scan request queue in step 18. Thus, individual scan requests and not chunks of plural scan requests are placed on the virus scan request queue of Smithson. See Smithson Col. 4 line 50 to Col. 5 line 5. Moreover, the "Time Requested" stamps in FIG. 3 are an indication that the plurality of requests shown in the virus scan request queue of FIG. 3 were placed onto the virus scan requests individually at different respective times.

Page 2 of the final Official Action says: "While Smithson does teaches individual requests, there is noting that precludes a plurality of individual requests being placed together (in particular if the requestor has priority)." (Emphasis added.) In reply, appellant respectfully points out that the test for inherency to satisfy anticipation is not whether a disclosed apparatus

or process is precluded from being operated or carried out in a more specific way as claimed by appellant. Instead, to show inherency to satisfy anticipation, the disclosed apparatus or process must necessarily perform in the specific way as claimed by appellant.

In order to be inherent, the nondisclosed element must be inevitable. In other words, it is not sufficient if the element is sometimes present and sometimes absent; it must be inevitably present. See, for example, Tyler Refrigeration v. Kysor Indus. Corp., 777 F.2d 687, 689, 227 U.S.P.Q. 845, 846-47 (Fed. Cir. 1985)(it is inherent that a claimed embodiment of a reference patent maintains an air curtain during a defrost cycle; the judge decided differently from the PTO with due consideration of the presumption of validity); W.L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 1548, 220 U.S.P.Q. 303, 309 (Fed. Cir. 1983), cert. denied sub nom., 469 U.S. 851 (1984), 105 S. Ct. 172 (1984)(process was inherently carried out by operation of a machine, and it was irrelevant that persons using the machine did not appreciate the results). See also MPEP 2112.02 ("Under the principles of inherency, if a prior art device, in its normal and usual operation, would necessarily perform the method claimed, then the method claimed will be considered to be anticipated by the prior art device.")

In short, it is not sufficient to simply consider multiple on-demand requests that happen to be on the queue of Smithson as a chunk. The fact that multiple requests of the same type or priority might be found together on Smithson's queue at any given time does not necessarily imply that these multiple requests were processed in the specific fashion recited in appellant's claim 6 by "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto

the virus scan request queue." (Emphasis added.) For the same reasons, McAfee also does not disclose "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue." (Emphasis added.)

Page 2 of the Advisory Action dated Nov. 3, 2009 says: "As long as there are a plurality of on-demand scan requests together that were specifically placed on the scan queue for any reason, the limitations of claim 6 are covered as there are no further explanations of how the grouping is done." Appellant respectfully disagrees, because "placing the chunks onto the virus scan request queue" is different and separate from "grouping the on-demand antivirus scan requests into chunks" and different from placing individual on-demand antivirus scan requests onto the virus scan request queue. How the grouping is done is different and separate from how on-demand anti-virus requests are placed on the virus scan request queue. One cannot simply presume that a plurality of one-demand scan requests seen together on the virus scan request queue must have arrived at that state by a process of "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue."

3. Claims 11-15 and 22-28 are patentable under 35 U.S.C. 102(e) and are not anticipated by McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002.

On pages 6-8 of the final Official Action, appellant's claims 11-15 and 22-28 were rejected under 35 U.S.C. 102(e) as being anticipated by McAfee ("GroupShield and the Microsoft Virus Scanning API"). Appellant respectfully disagrees.

McAfee describes a virus scanning API 2.0 said to have been released as an upgrade to Microsoft Exchange 2000 in Service Pack 1 to provide a number of enhancements including priority based queuing, multithreaded queue processing, and enhanced background scanning. As a client attempts to gain access to an Exchange item in the Exchange Server, a comparison is made to ensure that the message body and attachment (if present) has been scanned by the current virus signature file. If the content has not been scanned by the current virus signature file, then the corresponding item is submitted to GroupShield for scanning before that item is released to the client. In virus scanning API 2.0, a single queue processes all of the message body and attachment data. On-access requests are submitted as high-priority items. This queue is now serviced by a series of threads (the default number of threads is 2* number_of_processors + 1), with high-priority items always taking precedence. (McAfee, page 3.) Virus scanning API 2.0 also includes on-demand proactive scanning of messages. Items are submitted to a common information store queue as they are submitted to the information store. Each of these items receives a low priority in the queue, so that these items do not interfere with the scanning of the high-priority items. When all of the high-priority items have been scanned, virus scanning API 2.0 begins to scan low-priority items. The priority of items is dynamically upgraded to high priority if a client attempts to access the item when the item is in the low-priority queue. A

maximum of 30 items can exist at one time in the low priority queue, which is determined on a first in, first out basis. (McAfee, page 4.)

Claim 11

Claim 11 recites "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers; ..." Claim 11 also recites "inhibiting the distribution of the multiple ones of the on-demand anti-virus scan requests from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks." Appellant also does not see where McAfee discloses "inhibiting the distribution of multiple ones of the on-demand anti-virus scan requests from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" as recited in appellant's claim 11. Instead, in McAfee, if an on-demand request is next in line in McAfee's Global Scanning Queue and a scanner thread has just finished using a processor to satisfy an anti-virus scan request, the scanner thread would then begin scanning the on-demand request that is next in line in McAfee's Global Scanning Queue.

Claims 12, 13, and 14

Appellant respectfully submits that McAfee also does not disclose "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the

on-demand anti-virus scan requests, and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue" as recited in appellant's claim 12. Instead, in the proactive scanning of McAfee, when an item is submitted to the information store, an on-demand scan request of the item is placed on the low-priority queue so long as the queue is not full (a maximum of 30 items can exist at one time in the low-priority queue). If an on-demand scan request of the item is not placed on the low-priority queue when an item is submitted to the information store because the queue is full at that time, the item still could be scanned later as a result of background scanning or on-access scanning. In short, in McAfee, individual scan requests and not chunks of plural scan requests are placed on the queue so long as the queue is not full.

Claim 22

Among other things, appellant's claim 22 recites "wherein said at least one of the processors is programmed for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue."

With respect to "chunks" in McAfee, page 7 of the final Official Action says: "Page 3 [of McAfee] shows on-access scan requests being placed within 'chunks' of on-demand scan requests." Appellant respectfully disagrees. Page 3 of McAfee shows (under the heading

"Global Scanning Queue") a first box in which "Unscanned items are all placed in the queue with the same priority."; a second box in which "However, if a user accesses an item, it attains a high priority and jumps to the front of the queue."; and a third box in which "If a user saves an item to a folder, it is given a low priority." (Page 4 further explains that the on-demand requests with the same low priority are maintained as a first-in first-out queue.) Therefore, in appellant's view, the first box on page 3 of McAfee shows the low-priority queue of on-demand scan requests in the Global scanning queue, and the second box shows a scan request being moved from the low-priority queue to the front of the Global Scanning Queue when the scan request is promoted from an on-demand scan request to an on-access scan request. Therefore, appellant does not see "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue" (emphasis added) as recited in appellant's claims 6 and 22.

Claims 24, 25, and 26

Claim 24 recites "the file server further being programmed for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue." (Emphasis added.) Thus, claim 24 is distinguished from McAfee for the same reasons given above with respect to claim 22, and claim 24 further distinguishes McAfee for specifying how the chunks are placed on the queue; the chunks are placed consecutively onto the

virus scan request queue. It is not sufficient to simply consider multiple on-demand requests that happen to be on the queue of McAfee as a chunk.

Claims 15 and 28

Appellant's dependent claims 15 and 28 further recite "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks."

Appellant respectfully submits that McAfee does not disclose "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" as recited in appellant's claims 15 and 28. Appellant respectfully submits that it is unreasonable to interpret "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" as simply not placing an individual on-demand scan request on the queue of McAfee if the queue is full. For example, if the low-priority queue of McAfee contains a plurality of scan requests and this plurality of scan requests is considered a chunk, then in McAfee there is no inhibiting placement of a new on-demand scan request on the low-priority queue of McAfee until the low-priority queue is empty. Instead, so long as the queue is not full, a new on-demand scan request would be placed on the queue of McAfee.

Moreover, the definition of "inhibit" as "limit, block, or decrease the action or function of" as given on page 3 of the final Official Action is of no consequence because this definition

provides no justification for ignoring the claim limitation of "until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks." The limitation of "until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" is different from the queue being full or not.

Claim 23

Appellant's dependent claim 23 also further recites "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks."

As discussed above with respect to appellants claims 15 and 28, appellant respectfully submits that McAfee does not disclose "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" as recited in appellant's dependent claim 23. Appellant respectfully submits that it is unreasonable to interpret "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks" as simply not placing an individual on-demand scan request on the queue of McAfee if the queue is full.

Claim 27

Appellant's dependent claim 27 further recites "the file server is programmed for checking for each chunk whether the number of anti-virus scan requests on the virus checking

queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue." For the reasons discussed above with respect to claims 12, 13, and 14, appellant respectfully submits that McAfee does not disclose checking for each chunk whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue. In short, in McAfee, individual scan requests and not chunks of plural scan requests are placed on the queue so long as the queue is not full.

4. Claims 7, 12, and 24 are patentable under 35 U.S.C. 103(a) and are not obvious from Smithson et al. U.S. 6,802,012 in view of McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002.

On pages 9-10 of the Official Action, claims 7, 12, and 24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Smithson in view of McAfee. Appellant respectfully disagrees.

Claim 7

Appellant's dependent claim 7 further recites "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks." As discussed above, Smithson and McAfee place individual scan requests on a virus scan request queue, instead of "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue." In addition, inhibiting the placement of an individual scan request upon a virus scan request queue when the queue is full is different from "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks." Inhibiting the placement of a chunk upon the queue is different from inhibiting the placement of an individual scan request upon the queue, and the condition of the queue being full or not is different from the condition of completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

Claim 12

Appellant's claim 12 recites, among other things, "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each

chunk on the virus scan request queue." Thus, claim 12 is distinguished from Smithson and McAfee as discussed above because neither Smithson nor McAfee discloses "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and ... placing said each chunk on the virus scan request queue." (Emphasis added.) In addition, neither Smithson nor McAfee discloses "for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue" (emphasis added) because, as discussed above, inhibiting the placement of a chunk upon the queue is different from inhibiting the placement of an individual scan request upon the queue.

Claim 24

Appellant's claim 24 recites, among other things, "for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue." Thus, claim 24 is distinguished from Smithson and McAfee as discussed above because neither Smithson nor McAfee discloses "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and ... placing the chunks onto the virus scan request queue." (Emphasis added.)

5. Claims 30 and 33 are patentable over McAfee, "Groupshield and the Microsoft Virus Scanning API," May 1, 2002, in view of in view of Edwards U.S. Patent 7,188,367.

On pages 10-11 of the final Official Action, claims 30 and 33 were rejected under 35 U.S.C. 103(a) as being unpatentable over McAfee in view of Edwards (U.S. 7,188,367). Appellant respectfully disagrees. Edwards is cited for placing scan requests in a priority queue based on "optimal virus scanner throughput" (See Column 8 lines 5-21) and for teaching a threshold (Column 5, lines 65-67, Column 5, lines 1-6.) However, Edwards does not disclose the limitations of the base claims 12 and 27 that are missing from McAfee, as discussed above. As shown in Edwards FIG. 3, step 350, Edwards also places an individual scan request in a priority queue. (For example, compare Edwards FIG. 3 to Smithson FIG. 2.) In addition, the threshold of Edwards, Column 5, lines 65-67, and Column 5, lines 1-6, relates to "using the scan request's user characteristics, a pending scan request from user A may be determined to be more suitable than a pending scan request from user B if three of the four scanner threads are already scanning san requests from user B." It is respectfully submitted that improper hindsight would be required for the scan request threshold of Edwards to provide sufficient motivation for arriving at the appellant's chunk threshold of one-half of the chunk size for placing a chunk on the virus scan queue.

When determining whether a claim is obvious, an examiner must make "a searching comparison of the claimed invention – including all its limitations – with the teaching of the

prior art." In re Ochiai, 71 F.3d 1565, 1572 (Fed. Cir. 1995) (emphasis added). Thus, "obviousness requires a suggestion of all limitations in a claim." CFMT, Inc. v. Yieldup Intern. Corp., 349 F.3d 1333, 1342 (Fed. Cir. 2003) (citing In re Royka, 490 F.2d 981, 985 (CCPA 1974)). Moreover, as the Supreme Court stated, "there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." KSR Int'l v. Teleflex Inc., 127 S. Ct. 1727, 1741 (2007) (quoting In re Kahn, 441 F.3d 977, 988 (Fed. Cir. 2006) (emphasis added)). A fact finder should be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. See Id., 127 S. Ct. at 1742, citing Graham, 383 U. S. at 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "guard against slipping into the use of hindsight.").

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988). Neither Smithson nor McAfee recognizes that there is a problem with servicing a virus scan request queue that could be or should be solved by the appellant's chunk placement feature or appellant's chunk placement inhibiting feature. As discussed above, by managing on-demand virus scan requests in chunks of plural on-demand requests, the appellant's invention can keep multiple virus checkers busy scanning files in a file system without substantially reducing the availability of the virus checkers for on-access virus checking. This novel advantage of appellant's invention is objective evidence of non-obviousness.

In view of the above, the rejection of claims 6, 7, 11-15, 22-28, 30, and 33 should be reversed.

Respectfully submitted,

/ Richard C. Auchterlonie /

Richard C. Auchterlonie
Reg. No. 30,607
NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
713-571-3400 (telephone)
713-456-2836 (telefax)

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

6. A method of operating a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, the method comprising:

combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue; and

distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers;

which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue.

7. The method as claimed in claim 6, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

11. A method of operating a plurality of virus checkers, the method comprising:

distributing on-demand anti-virus scan requests and on-access anti-virus scan requests to the virus checkers so that the virus checkers perform on-demand anti-virus scanning concurrent

with on-access anti-virus scanning;

which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers; and

which includes inhibiting the distribution of the multiple ones of the on-demand anti-virus scan requests from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

12. A method of operating a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, the method comprising:

combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue; and

a pool of threads distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads,

which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than

a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue.

13. The method as claimed in claim 12, wherein the on-access anti-virus scan requests are produced in response to user access to files.

14. The method as claimed in claim 12, wherein the on-demand anti-virus scan requests are produced in response to a system administrator requesting a scan of files within a specified file system.

15. The method as claimed in claim 12, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

22. A virus checking system comprising:

a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning;

a virus scan request queue; and

at least one processor coupled to the virus checkers and the virus scan request queue for sending virus scan requests from the virus scan request queue to the virus checkers, said at least one processor being programmed for placing on-demand anti-virus scan requests and on-access anti-

virus scan requests onto the virus scan request queue, and for distributing the on-demand anti-virus scan requests and the on-access virus scan requests from the virus scan request queue to the virus checkers;

wherein said at least one of the processors is programmed for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue.

23. The virus checking system as claimed in claim 22, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

24. A virus checking system comprising:

a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning; and

a file server coupled to the virus checkers for sending virus scan requests to the virus checkers, the file server including a virus scan request queue, and the file server being programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue; and for executing multiple threads for distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads, the file server

further being programmed for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue.

25. The virus checking system as claimed in claim 24, wherein the file server is programmed for producing the on-access anti-virus scan requests in response to user access to files.

26. The virus checking system as claimed in claim 24, wherein the file server is programmed to produce the on-demand anti-virus scan requests in response to a system administrator requesting a scan of files within a specified file system.

27. The virus checking system as claimed in claim 24, wherein the file server is programmed for checking for each chunk whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue.

28. The virus checking system as claimed in claim 24, wherein the file server is programmed for inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

30. The method as claimed in claim 12, wherein the chunks have a size equal to a certain number of on-demand anti-virus scan requests, and the threshold is one-half of the size.

33. The virus checking system as claimed in claim 27, wherein the chunks have a size equal to a certain number of on-demand anti-virus scan requests, and the threshold is one-half of the size.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.